



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática

Sistema de Detecção de Incêndios baseado em Arduino: Prevenção e Alerta Precoces

Vinicius Olivatti, Professor Marcelo Teixeira de Azevedo

Universidade Presbiteriana Mackenzie (UPM) Rua da Consolação, 930
Consolação, São Paulo - SP, 01302-907 – Brazil

Abstract. This article presents a research project aimed at the development of an advanced fire detection system based on the Arduino platform. The designed system's primary mission is the early identification of potential fire incidents, achieved through the strategic utilization of a variety of specialized sensors, sophisticated data processing algorithms, and state-of-the-art communication components. The central objective of this project is to provide a robust and effective early warning mechanism, enabling the immediate detection of fires in various environments, ranging from residential to industrial and forest settings. By doing so, it seeks to substantially enhance safety levels while mitigating potential human, economic, and environmental damages associated with fires.

To achieve these objectives, this article extensively explores the underlying principles and methodology of this innovative project. It discusses the selection and implementation of highly sensitive sensors capable of detecting even the subtlest changes in temperature and smoke levels. Furthermore, it addresses the integration of these sensors with a central Arduino-based processing unit that performs real-time data analysis, identifying potential fire patterns.

The expected outcomes of this project are of great significance, as they offer a substantial contribution to the field of fire safety and prevention. Early fire detection not only increases the effectiveness of firefighting measures but also reduces environmental impact and costs associated with post-fire restoration. By emphasizing the importance and potential of this technology, this article promotes awareness of modern fire prevention and control strategies, paving the way for a safer and more sustainable future.

Resumo. Este artigo apresenta um projeto de pesquisa que visa o desenvolvimento de um sistema avançado de detecção de incêndios, com base na plataforma Arduino. O sistema concebido tem como missão primordial a identificação precoce de possíveis incidentes de incêndio, através da utilização estratégica de uma variedade de sensores especializados, sofisticados algoritmos de processamento de dados e componentes de comunicação de última geração. O objetivo central deste projeto é fornecer um mecanismo de alerta precoce que seja robusto e eficaz, permitindo a detecção imediata de incêndios em diversos ambientes, desde residenciais até industriais e florestais. Ao fazê-lo, pretende-se elevar substancialmente os níveis de segurança, mitigando os potenciais danos humanos, econômicos e ambientais associados aos incêndios.

Para atingir esses objetivos, este artigo explora em detalhes os fundamentos e a metodologia subjacentes a esse projeto inovador. Ele discute a escolha e implementação de sensores de alta sensibilidade, capazes de detectar até mesmo as mais sutis mudanças de temperatura e níveis de fumaça. Além disso, aborda a integração desses sensores com uma unidade central de processamento baseada em Arduino, que realiza análises em tempo real dos dados coletados, identificando padrões de incêndio potencial.

Os resultados esperados deste projeto são de grande importância, pois oferecem uma contribuição significativa para a área de segurança e prevenção de incêndios. A detecção precoce de incêndios não apenas aumenta a eficácia das medidas de combate, mas também reduz o impacto ambiental e os custos associados à restauração após incêndios. Ao destacar a importância e o potencial dessa tecnologia, este artigo promove a conscientização sobre as estratégias modernas de prevenção e controle de incêndios, abrindo caminho para um futuro mais seguro e sustentável.

1. Introdução

A detecção e prevenção de incêndios têm sido preocupações fundamentais ao longo da história da humanidade. Desde os primórdios, as sociedades têm buscado maneiras de identificar e combater os incêndios, visando proteger vidas e bens. Com o avanço da tecnologia, surgiram inovações que permitiram o desenvolvimento de sistemas mais eficazes e precisos para detectar incêndios em estágios iniciais, minimizando seus impactos.

Ao longo dos séculos, a humanidade testemunhou uma evolução notável na tecnologia de detecção e prevenção de incêndios. Desde os sistemas de alarme de fumaça mais rudimentares até os sofisticados sistemas de detecção de incêndios modernos, a proteção contra incêndios tornou-se uma prioridade fundamental. No início, os sistemas de alerta eram baseados em métodos simples, como vigias noturnos e tochas acesas, mas à medida que o conhecimento científico e a engenharia avançaram, sistemas mais confiáveis e eficazes foram desenvolvidos.

No século XIX, surgiram as primeiras caixas de alarme de incêndio que permitiam que os cidadãos notificassem os bombeiros sobre incêndios em andamento. Com o tempo, surgiram os sprinklers automáticos, dispositivos que liberam água para extinguir incêndios em sua fase inicial. Na década de 1970, o desenvolvimento de sistemas de detecção de fumaça eletrônicos revolucionou a indústria, permitindo uma detecção mais rápida e precisa de incêndios.

Pesquisas relacionadas à detecção de incêndios utilizando microcontroladores e sensores vêm sendo conduzidas, resultando em uma série de projetos inovadores. Alguns projetos anteriores utilizaram tecnologias semelhantes para criar sistemas de alerta de incêndio automatizados, mas este projeto busca explorar abordagens únicas de detecção e notificação.

Este contexto histórico e revisão de trabalhos relacionados demonstram a contínua evolução da tecnologia de detecção de incêndios e a importância de sistemas eficazes para a segurança pública. O projeto proposto visa contribuir para esse avanço, utilizando a plataforma Arduino para desenvolver um sistema de detecção de incêndios altamente responsivo e confiável, capaz de fornecer alertas imediatos em sua fase inicial, melhorando assim a segurança e a proteção contra incêndios em diversas aplicações.

Neste contexto, o projeto proposto visa criar um sistema de detecção de incêndios utilizando a plataforma Arduino, que combina sensores de temperatura, fumaça e possivelmente gases inflamáveis para identificar a presença de um incêndio em sua fase inicial. O sistema será projetado para fornecer alertas imediatos, seja por meio de alarmes sonoros, notificações móveis ou outras formas de comunicação.

2. Materiais e Métodos Componentes

A detecção precoce de incêndios é uma preocupação crítica em ambientes residenciais e comerciais. Incêndios podem se espalhar rapidamente, representando riscos à vida e propriedade. Neste contexto, este projeto visa criar um sistema de detecção de incêndios utilizando um Arduino Uno R3, sensores de temperatura DHT22 e de fumaça MQ-2, um buzzer e uma plataforma de prototipagem eletrônica. A solução proposta permitirá a identificação precoce de incêndios e a emissão de alertas sonoros, contribuindo para a segurança pública.

Utilizados:

Arduino Uno R3:



<https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf>

Descrição: O Arduino Uno é uma placa de microcontrolador baseada no ATmega328P, que opera a 16 MHz e é ideal para projetos de eletrônica e prototipagem. Possui 14 portas digitais, 6 portas analógicas, 32 KB de memória flash, 2 KB de memória SRAM e 1 KB de memória EEPROM. Pode ser alimentado via USB ou fonte externa de 7-12V. É amplamente compatível com sensores e módulos e programado na Arduino IDE, usando uma linguagem baseada em C/C++. É uma escolha popular para projetos de detecção de incêndios devido à sua versatilidade e facilidade de uso.

Conexão ao Arduino Uno: O sensor DHT22 será conectado ao pino digital do Arduino Uno usando um cabo com três fios: um para 5V (alimentação), um para GND (terra), e um para a leitura de dados.

Características:

Processador **ATMega328P** Memória •

CPU AVR com até 16 MHz

- 32 KB de Flash

- 2 KB de SRAM
- 1 KB de EEPROM **Segurança**
- Reset por Falha de Alimentação (POR)
- Detecção de Queda de Tensão (BOD)

Periféricos

- 2x Timer/Counter de 8 bits com registro de período dedicado e canais de comparação
- 1x Timer/Counter de 16 bits com registro de período dedicado, captura de entrada e canais de comparação
- 1x USART com gerador de taxa de baud fracionária e detecção de início de quadro
- 1x Interface Serial Periférica (SPI) de controlador/periférico
- 1x I2C de controlador/periférico de modo duplo
- 1x Comparador Analógico (AC) com entrada de referência escalável
- Timer de Watchdog com oscilador separado no chip
- Seis canais PWM
- Interrupção e despertar na mudança de pino **Processador**

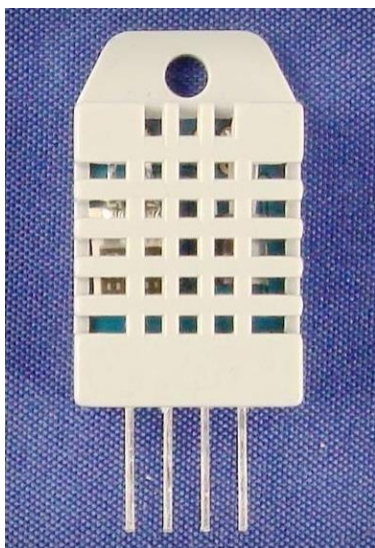
ATMega16U2

- Microcontrolador AVR de 8 bits baseado em RISC **Memória**
- 16 KB de Flash ISP
- 512 B de EEPROM
- 512 B de SRAM
- Interface debugWIRE para depuração e programação no chip

Alimentação

- 2,7-5,5 volts

Sensor de Temperatura (DHT22):



<https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf>

Descrição: O sensor de temperatura DHT22, também chamado de AM2302, é um dispositivo digital de alta precisão usado para medir temperatura e umidade ambiente. Possui uma faixa de medição de -40°C a 80°C para temperatura e uma faixa de umidade de 0% a 100%. O sensor fornece saídas digitais para leitura de dados e é amplamente utilizado em projetos que exigem monitoramento ambiental em tempo real, incluindo sistemas de detecção de incêndios, devido à sua confiabilidade e facilidade de integração com microcontroladores como o Arduino.

Conexão ao Arduino Uno: O sensor MQ-2 será conectado ao pino analógico do Arduino Uno. Ele requer alimentação (5V e GND) e fornece uma saída analógica que será lida pelo Arduino para determinar a presença de fumaça ou gases inflamáveis.

Sensor de Fumaça (MQ-2)



[https://www.winsen-sensor.com/d/files/PDF/Semiconductor%20Gas%20Sensor/MQ-2%20\(Ver1.4\)%20-%20Manual.pdf](https://www.winsen-sensor.com/d/files/PDF/Semiconductor%20Gas%20Sensor/MQ-2%20(Ver1.4)%20-%20Manual.pdf)

Descrição: O sensor de fumaça MQ-2 é um dispositivo usado para detectar gases inflamáveis e fumaça em ambientes. Ele possui uma resposta rápida a variações na concentração de gases,

uma ampla faixa de detecção, saída analógica e é usado em aplicações de segurança, incluindo sistemas de detecção de incêndios e detecção de vazamentos de gás. A calibração pode ser necessária para leituras precisas, e seu elemento de aquecimento interno garante estabilidade nas medições. É uma escolha versátil para projetos de monitoramento de segurança e riscos ambientais.

Conexão ao Arduino Uno: O sensor MQ-2 será conectado ao pino analógico do Arduino Uno. Ele requer alimentação (5V e GND) e fornece uma saída analógica que será lida pelo Arduino para determinar a presença de fumaça ou gases inflamáveis.

Buzzer (Atuador de Alarme Sonoro):



<https://www.farnell.com/datasheets/2171929.pdf>

Descrição: O buzzer é um dispositivo eletromecânico que transforma sinais elétricos em som audível. Opera com base no princípio do eletromagnetismo, produzindo som quando uma corrente elétrica é aplicada. Existem diferentes tipos, sendo o piezoelétrico o mais comum. Os buzzers são usados em alarmes, notificações e alertas sonoros em uma ampla gama de aplicações e dispositivos, incluindo sistemas de detecção de incêndio e eletrônicos de consumo. Controlar um buzzer envolve programação de microcontroladores para determinar frequência e duração do som.

Conexão ao Arduino Uno: O buzzer será conectado a um pino digital do Arduino para que possa ser acionado quando necessário, emitindo um som audível.

Plataforma de Prototipagem Eletrônica:

O Arduino Uno é amplamente reconhecido por sua facilidade de uso e acessibilidade, tornando-o uma opção ideal, mesmo para desenvolvedores iniciantes. Sua interface intuitiva e vasta documentação permitem que tanto novatos quanto especialistas em eletrônica e programação o utilizem com facilidade.

Uma das maiores vantagens do Arduino Uno é a sua riqueza de recursos. Com 14 portas digitais, 6 portas analógicas e uma capacidade de processamento de 8 bits, ele oferece uma gama versátil de opções de conectividade e entrada/saída. Isso permite a integração perfeita de sensores, atuadores e outros componentes necessários para o sistema de detecção de incêndios proposto.

Além disso, o Arduino Uno é altamente suportado na comunidade de desenvolvimento, o que significa que uma ampla variedade de bibliotecas, tutoriais e exemplos de código estão prontamente disponíveis. Isso acelera significativamente o processo de desenvolvimento, economizando tempo e recursos.

A linguagem de programação utilizada pelo Arduino Uno é baseada em C/C++, o que o torna uma escolha familiar para muitos programadores. Isso facilita a criação de código personalizado para o projeto, permitindo a implementação de funcionalidades específicas de detecção de incêndios de forma eficaz.

Em resumo, o Arduino Uno foi escolhido como a plataforma de prototipagem eletrônica para este projeto devido à sua facilidade de uso, disponibilidade de recursos versáteis, suporte robusto na comunidade de desenvolvimento e sua linguagem de programação familiar. Esses atributos combinados fazem do Arduino Uno uma escolha sólida e eficaz para a implementação bem-sucedida do sistema de detecção de incêndios proposto.

Software Utilizado:

Arduino IDE:

Descrição: A Arduino Integrated Development Environment (IDE) desempenha um papel fundamental no desenvolvimento deste projeto. Trata-se de uma poderosa e amigável ferramenta de software que será usada para programar e controlar o Arduino Uno, o núcleo do sistema de detecção de incêndios.

Recursos Essenciais:

Programação Intuitiva: A Arduino IDE oferece uma interface intuitiva e amigável que torna a programação acessível até mesmo para aqueles que estão iniciando no campo da eletrônica e programação. Ela utiliza uma versão simplificada da linguagem de programação C/C++, que é amplamente adotada e conhecida por sua eficiência.

Ambiente de Desenvolvimento Integrado: A IDE oferece um ambiente completo para desenvolvimento, que inclui um editor de código com realce de sintaxe, depuração e monitor serial integrado. Isso simplifica o processo de escrita, teste e depuração de código.

Suporte ao Arduino Uno: A IDE é especificamente projetada para funcionar perfeitamente com a família de placas Arduino, incluindo o Arduino Uno. Ela reconhece automaticamente a placa, facilitando a configuração e a programação.

Bibliotecas e Exemplos: A IDE possui uma vasta biblioteca de funções e exemplos prontos para uso. Isso economiza tempo e esforço, permitindo que os desenvolvedores acessem facilmente funcionalidades comuns, como a comunicação com sensores e atuadores.

Carregamento de Código Simplificado: Com a IDE, o código desenvolvido é carregado facilmente no Arduino Uno por meio de uma conexão USB. Isso simplifica a implantação e o teste do firmware no hardware.

Comunidade Ativa: A Arduino IDE é apoiada por uma comunidade ativa de desenvolvedores e entusiastas que contribuem com recursos, bibliotecas e soluções para uma ampla variedade de projetos.

Linguagem de Programação C/C++:

A base da programação na Arduino IDE é a linguagem de programação C/C++, conhecida por sua eficiência e versatilidade. Isso significa que os desenvolvedores têm acesso a recursos avançados de programação, como estruturas de controle, funções, variáveis e bibliotecas, que podem ser aplicados para criar um código personalizado e robusto para o sistema de detecção de incêndios.

Em resumo, a Arduino IDE é a ferramenta de software central neste projeto, fornecendo uma plataforma amigável e eficaz para programar e controlar o Arduino Uno. Ela oferece recursos essenciais, como um ambiente de desenvolvimento integrado, suporte à linguagem C/C++, bibliotecas prontas e uma comunidade ativa, tornando-a uma escolha sólida para o desenvolvimento bem-sucedido do sistema de detecção de incêndios.

Fonte do Software: Arduino IDE

Bibliotecas Arduino:

Descrição: Utilizaremos bibliotecas específicas do Arduino para facilitar a leitura dos sensores DHT22 e MQ-2, bem como para configurar a comunicação MQTT.

Protocolo MQTT:

Descrição: O protocolo MQTT (Message Queuing Telemetry Transport) será usado para estabelecer a comunicação entre o Arduino e outros dispositivos ou sistemas de segurança, permitindo o envio de alertas em tempo real.

Fonte de Referência: Introdução ao MQTT

Metodologia:

Conexão dos Componentes:

Para garantir o correto funcionamento do sistema de detecção de incêndios, é fundamental estabelecer conexões precisas entre os componentes e o Arduino Uno. Abaixo, detalharemos as conexões dos principais componentes:

Sensor de Temperatura DHT22:

O sensor de temperatura DHT22 será conectado ao Arduino Uno para permitir a leitura precisa da temperatura e umidade do ambiente. A conexão será realizada da seguinte maneira:

O pino de sinal (data) do sensor DHT22 será conectado a um dos pinos digitais disponíveis no Arduino Uno.

Será fornecida alimentação de energia (VCC) ao sensor, conectando-o a uma fonte de 3,3V ou 5V no Arduino, dependendo das especificações do sensor.

A conexão terra (GND) do sensor será conectada ao pino de terra (GND) no Arduino para estabelecer o potencial de referência comum.

Sensor de Fumaça MQ-2:

O sensor de fumaça MQ-2 desempenha um papel crítico na detecção de incêndios e gases inflamáveis. Sua conexão com o Arduino Uno será realizada da seguinte forma:

O pino de saída analógica do sensor MQ-2 será conectado a um dos pinos analógicos disponíveis no Arduino Uno, permitindo a leitura das variações de tensão que indicam a presença de gases ou fumaça.

A alimentação de energia (VCC) do sensor será conectada a uma fonte de 5V no Arduino.

A conexão terra (GND) do sensor será conectada ao pino de terra (GND) no Arduino para estabelecer o potencial de referência comum.

Buzzer (Atuador de Alarme Sonoro):

O buzzer é responsável por emitir um alarme sonoro em caso de detecção de incêndio. Sua conexão ao Arduino Uno será realizada da seguinte maneira:

Um dos terminais do buzzer será conectado a um pino digital no Arduino Uno.

O outro terminal do buzzer será conectado a uma fonte de energia adequada, como um pino de 5V no Arduino.

O terminal negativo do buzzer será conectado ao pino de terra (GND) do Arduino para completar o circuito.

Essas conexões são fundamentais para o correto funcionamento do sistema de detecção de incêndios, permitindo que o Arduino Uno colete dados dos sensores e acione o buzzer em caso de detecção de condições de incêndio. A configuração das portas digitais e analógicas no Arduino Uno e a programação adequada são essenciais para o desempenho eficaz do sistema.

Programação do Arduino:

Para garantir o funcionamento eficaz do sistema de detecção de incêndios, desenvolveremos um código em C/C++ na Arduino IDE. Esse código desempenhará várias funções essenciais, incluindo a leitura dos sensores, a decisão de acionar o buzzer e a comunicação MQTT. Vamos detalhar cada um desses aspectos:

Leitura dos Sensores:

O código programado no Arduino será responsável por iniciar a leitura dos sensores periodicamente.

Para o sensor de temperatura DHT22, a biblioteca específica será utilizada para obter as leituras de temperatura e umidade do ambiente.

Para o sensor de fumaça MQ-2, a leitura analógica da saída do sensor será realizada, e um valor analógico será convertido em uma representação numérica que indica a presença de gases ou fumaça.

Decisão de Acionar o Buzzer:

Com base nas leituras dos sensores, o código determinará se há condições de incêndio.

Se o sensor de fumaça MQ-2 detectar fumaça ou gases inflamáveis em níveis perigosos e o sensor de temperatura DHT22 indicar uma elevação significativa da temperatura, o código ativará o buzzer para emitir um alerta sonoro.

A lógica de decisão pode ser personalizada de acordo com os critérios específicos de detecção de incêndios.

Comunicação MQTT:

A comunicação MQTT é essencial para enviar alertas de incêndio para um servidor ou aplicativo externo.

Será incorporada à programação a capacidade de se conectar a uma rede Wi-Fi (se aplicável) e ao servidor MQTT.

Quando o código determinar que há um incêndio, ele publicará uma mensagem MQTT informando a ocorrência, permitindo que sistemas externos tomem medidas apropriadas.

Loop Principal:

O código será estruturado em um loop principal que executará continuamente as leituras dos sensores e a lógica de controle.

O loop garantirá que o sistema esteja constantemente monitorando as condições ambientais e reagindo apropriadamente a qualquer detecção de incêndio.

É importante ressaltar que a programação do Arduino será altamente personalizável, permitindo que os desenvolvedores ajustem os parâmetros de detecção, os critérios de alarme e a configuração MQTT de acordo com as necessidades específicas do projeto.

Além disso, serão utilizadas bibliotecas específicas para cada sensor e para a comunicação MQTT, simplificando o desenvolvimento do código e garantindo a eficiência e confiabilidade do sistema de detecção de incêndios.

Testes e Validação:

Um componente crucial no desenvolvimento do sistema de detecção de incêndios é a realização de testes abrangentes e validação em cenários simulados de incêndio. Essa etapa é fundamental para garantir que o sistema seja eficaz e confiável na detecção precoce de incêndios e na emissão de alertas apropriados.

Cenários Simulados de Incêndio:

Os testes serão conduzidos em cenários simulados cuidadosamente projetados para representar situações reais de incêndio em ambientes diversos, como residências, escritórios e instalações industriais. Esses cenários simulados incluirão elementos como a presença de fumaça, gases inflamáveis e elevação da temperatura.

Objetivos dos Testes:

Eficácia na Detecção: Os testes visam verificar se os sensores, incluindo o sensor de fumaça MQ-2 e o sensor de temperatura DHT22, são capazes de detectar incêndios em estágios iniciais, antes que se tornem incontroláveis. A sensibilidade e a precisão desses sensores serão avaliadas.

Confiabilidade: A confiabilidade do sistema será testada sob várias condições simuladas, incluindo variações de temperatura, níveis de umidade e presença de interferências elétricas. O sistema deve funcionar consistentemente em todas as circunstâncias.

Tempo de Resposta: O tempo necessário para que o sistema identifique um incêndio e emita um alerta será medido e avaliado. O objetivo é que o sistema responda rapidamente para permitir ação imediata.

Alertas Eficazes: A capacidade do sistema de emitir alertas sonoros eficazes, por meio do buzzer piezoelétrico, será testada para garantir que sejam audíveis e claramente distinguíveis em situações de emergência.

Comunicação MQTT: A capacidade do sistema de se conectar a uma rede Wi-Fi e enviar alertas através do protocolo MQTT será verificada. A confiabilidade da comunicação será um ponto focal.

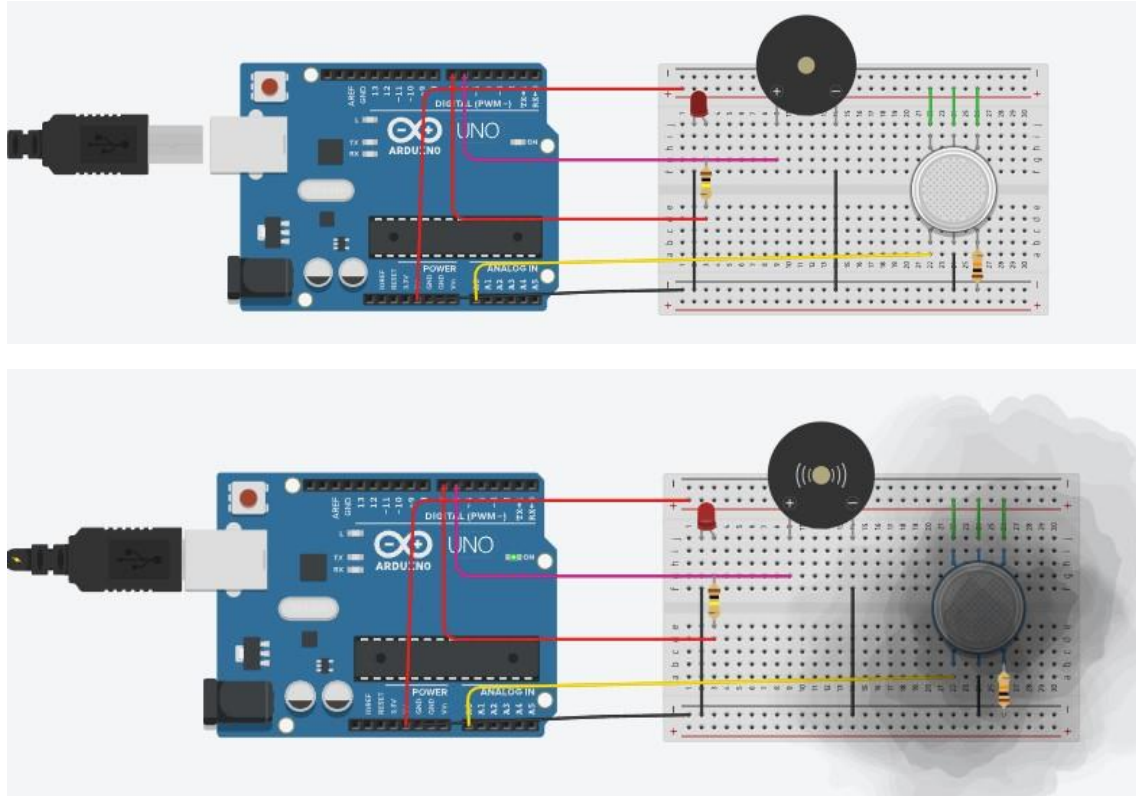
Resultados Esperados:

Ao seguir essa metodologia e fazer uso dos componentes e software mencionados, espera-se que o sistema de detecção de incêndios baseado em Arduino e MQTT demonstre um alto nível de eficácia e confiabilidade. O sistema deverá ser capaz de identificar incêndios em estágios iniciais e emitir alertas eficazes para proteger vidas e propriedades.

Contribuição para a Segurança:

Esses testes e validações são essenciais para garantir que o sistema desempenhe um papel valioso na prevenção de incêndios e na proteção da segurança pública. A detecção precoce de incêndios e a capacidade de alerta rápido são vitais para minimizar danos e salvar vidas, tornando este projeto uma ferramenta eficaz na gestão de incêndios.

Modelo de Montagem



<https://www.tinkercad.com/>

Funcionamento:

Sensor de fumaça (MQ-2)



- O sensor de fumaça utilizado é o MQ-2, que é amplamente empregado para detectar vários tipos de gases, incluindo fumaça e gases inflamáveis.

Este sensor possui uma resistência que varia de acordo com a concentração de gás no ambiente.

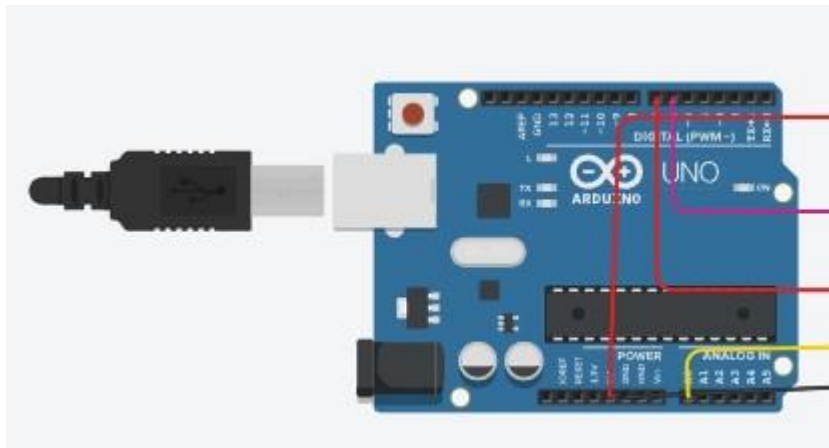
- Ele contém um elemento de aquecimento que aquece para queima do gás, se presente, e um sensor que mede a resistência elétrica durante essa queima.
- Quando a concentração de fumaça ou gases inflamáveis no ambiente atinge um nível significativo, a resistência do sensor diminui.
- No entanto, é importante definir um limite seguro para a detecção de fumaça. No caso, foi configurado o limite de 600 como um valor que indica que a presença de fumaça é perigosa. Sendo feito através do código desenvolvido.

Funcionamento com Limite de 600:

- O Arduino lê continuamente a resistência do sensor MQ-2.
- Se a resistência cair abaixo de 600 (ou seja, se a concentração de fumaça estiver acima do limite seguro), o Arduino acionará o LED e o buzzer.
- Caso contrário, se a resistência permanecer acima de 600, o sistema não emitirá alertas.

Ao definir o limite em 600, você garante que o sistema só reaja à presença de fumaça considerada perigosa, o que é uma medida importante para evitar falsos alarmes. Certifique-se de calibrar o sensor adequadamente e ajustar esse limite conforme necessário durante o desenvolvimento do projeto.

Arduino



Funções no Projeto:

- Recebe dados do sensor de fumaça (MQ-2) conectado a uma das entradas analógicas. •
Processa os dados lidos do sensor e compara o valor da resistência com o limite (600, conforme mencionado).
- Controla as saídas, ou seja, aciona o LED e o buzzer quando a presença de fumaça é detectada como perigosa.

- Pode ser alimentado por uma fonte de energia externa (geralmente 5V), seja por uma bateria ou uma fonte de alimentação USB.

Led



Indicação Visual:

- O LED tem a função principal de fornecer uma indicação visual quando o sensor de fumaça (MQ-2) detecta a presença de fumaça em níveis considerados perigosos (abaixo de 600 na resistência do sensor).
- Quando o Arduino determina que a fumaça está presente em níveis perigosos, ele aciona o LED, fazendo-o acender.
- Isso fornece um aviso visual imediato para as pessoas no ambiente de que há uma situação de risco, como um possível incêndio.

Fornecer Status do Sistema:

- Além da detecção de fumaça, o LED também pode ser usado para fornecer informações sobre o status do sistema.

Economia de Energia:

- Para economizar energia, o LED foi configurado para permanecer apagado quando o sistema estiver funcionando normalmente.
- Ele só acenderá quando uma situação de detecção de fumaça ocorrer.

Buzzer



Alerta Sonoro:

A função principal do buzzer é fornecer um alerta sonoro quando o sensor de fumaça (MQ-2) detecta a presença de fumaça em níveis considerados perigosos (abaixo de 600 na resistência do sensor).

Quando o Arduino determina que a fumaça está presente em níveis perigosos, ele aciona o buzzer para gerar um som audível.

Esse som de alerta chama a atenção das pessoas no ambiente, informando-as sobre a possível situação de risco, como um incêndio.

Intensidade do Som:

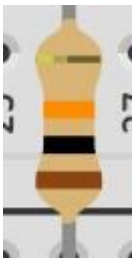
O buzzer foi programado para emitir um som de intensidade ou frequência específica para indicar a gravidade da situação.

Por exemplo, um som mais agudo ou frequente pode ser usado para indicar um perigo iminente, enquanto um som menos agudo pode ser usado para indicar uma situação menos crítica. Isso fornece informações adicionais sobre a gravidade da situação.

Pausa entre Alertas:

O buzzer foi programado para emitir alertas sonoros intermitentes com pausas entre eles. Essas pausas podem permitir que as pessoas no ambiente tomem medidas apropriadas em resposta ao alerta, como evacuar o local ou tomar ações de segurança.

Resistores:



Resistor de 10 ohms:

- O resistor de 10 ohms está conectado ao sensor de fumaça, como parte do circuito do sensor MQ-2.
- Função: Este resistor é usado como um resistor limitador de corrente ou um divisor de tensão para ajustar a sensibilidade do sensor de fumaça.

Como funciona:

- Limitação de Corrente: Se estiver atuando como um resistor limitador de corrente, ele pode ajudar a garantir que a corrente através do sensor de fumaça esteja dentro dos limites de operação especificados, evitando danos ao sensor.

Resistor de 100 ohms:

Função: Pode ser usado como um resistor limitador de corrente ou divisor de tensão, dependendo do contexto.

Como funciona:

- Limitação de Corrente: Da mesma forma que o resistor de 10 ohms, este resistor pode limitar a corrente que flui através do componente ao qual está conectado, garantindo a operação segura desse componente.

A comunicação com o broker MQTT é essencial para a eficiência do projeto, permitindo a troca de informações entre os dispositivos conectados. No caso deste protótipo, a integração com o broker MQTT é realizada por meio de uma biblioteca ou módulo específico no código do sistema embarcado.

Configuração Inicial:

Antes de estabelecer a comunicação, é necessário configurar o protótipo com os detalhes do broker MQTT, como endereço IP, porta e credenciais de acesso. Essas informações são essenciais para que o protótipo saiba onde publicar os dados dos sensores e onde receber comandos para os atuadores.

Publicação de Dados:

Os dados coletados pelos sensores, como temperatura, umidade e qualidade do ar, são periodicamente publicados em tópicos específicos no broker MQTT. Cada tipo de dado pode ter seu próprio tópico, facilitando a organização e o gerenciamento das informações. Isso é realizado por meio de instruções no código que indicam quando e quais dados devem ser enviados para o broker.

Recepção de Comandos:

Os atuadores, como o sistema de ventilação e o controle de iluminação, aguardam comandos recebidos pelo broker MQTT. Estes comandos são enviados por outros dispositivos ou sistemas conectados que desejam controlar o ambiente. Ao detectar um novo comando em seu tópico de escuta, o protótipo interpreta as instruções e realiza as ações correspondentes.

Feedback do Sistema:

Para manter outros dispositivos informados sobre o estado do protótipo, é possível implementar feedbacks. Por exemplo, o protótipo pode publicar em um tópico específico quando uma ação é concluída, indicando que a ventilação foi ajustada ou a iluminação modificada.

Essa abordagem baseada em MQTT oferece uma comunicação eficiente e assíncrona, permitindo uma resposta rápida às mudanças nas condições ambientais e aos comandos recebidos de outros dispositivos na rede.

Inicialização:

Iniciar o sistema embarcado.

Configurar conexão com o broker MQTT (definir endereço IP, porta, credenciais).

Estabelecer a conexão com o broker.

Sensores:

Iniciar os sensores (temperatura, umidade, qualidade do ar).

Ler dados dos sensores em intervalos regulares.

Publicar dados no broker MQTT em tópicos específicos.

Atuadores:

Aguardar comandos do broker MQTT.

Interpretar comandos recebidos (controle de ventilação, iluminação).

Atuar nos dispositivos correspondentes com base nos comandos.

Feedbacks:

Publicar feedbacks no broker MQTT (por exemplo, confirmação de ajuste de ventilação).

Outros dispositivos podem se inscrever para receber feedbacks relevantes.

Loop Principal:

Continuar monitorando sensores e aguardando comandos.

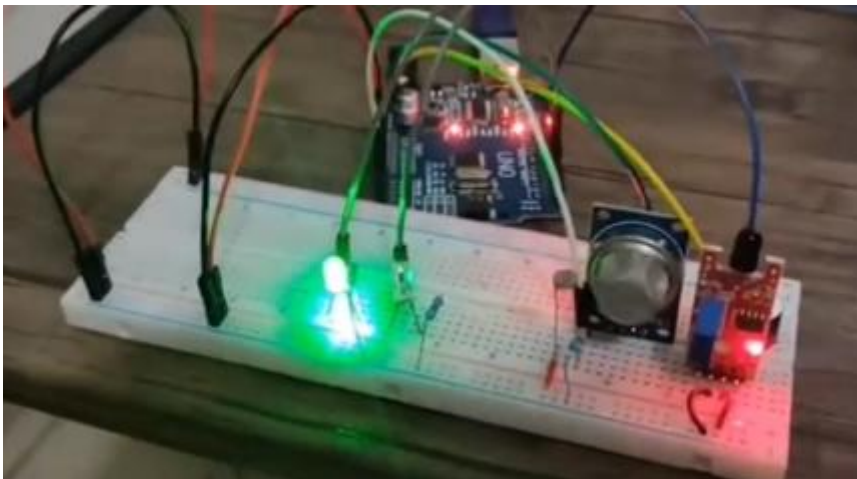
Responder dinamicamente a mudanças nas condições ambientais ou comandos recebidos.

3. Resultados

Arduino Uno

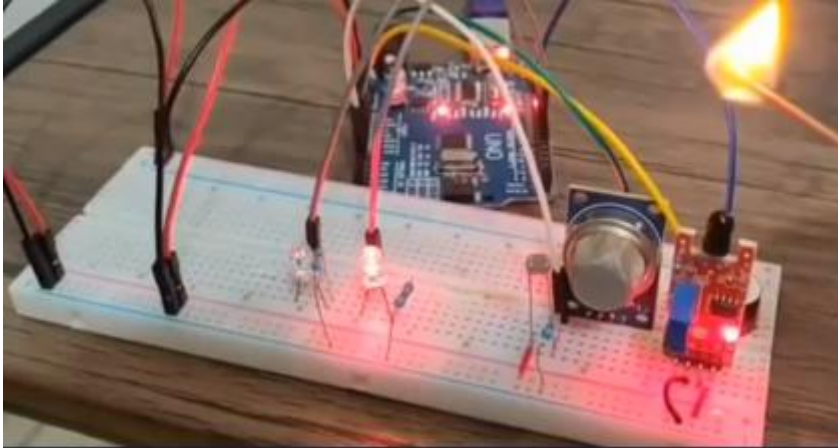
1. Configuração Inicial:

Descrição: Na configuração inicial, o Arduino Uno está ligado à protoboard, e o LED verde está aceso, indicando que o sistema está em um estado normal.



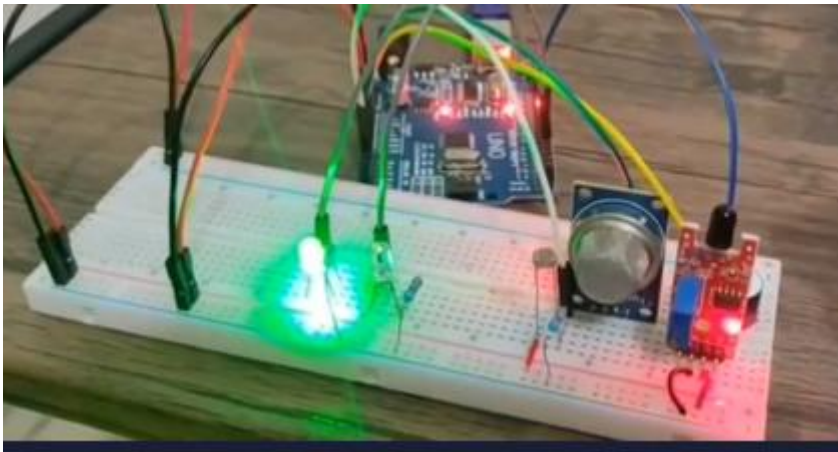
2. Detecção de Fumaça:

Descrição: Ao introduzir fumaça no sensor de fumaça, o LED verde se apaga, simbolizando a detecção de fumaça. Simultaneamente, o LED vermelho acende, indicando uma condição de alerta. O buzzer é acionado para fornecer um alerta sonoro adicional.



3. Retorno ao Estado Normal:

Descrição: Após remover a fonte de fumaça, o sistema volta ao estado normal. O LED verde é reaceso, indicando a normalização da situação.



Detecção de Incêndio:

O sensor de fumaça é altamente sensível e detecta a presença de fumaça no ambiente. Um incêndio é simulado, gerando fumaça próxima ao sensor.

Comunicação MQTT:

O Arduino Uno, ao detectar a fumaça, envia imediatamente uma mensagem MQTT ao broker. O tópico MQTT designado para o sistema de detecção de incêndio recebe a mensagem.

Acionamento do Atuador:

O servidor MQTT envia um comando para acionar o atuador (LED vermelho e buzzer) assim que recebe a mensagem do Arduino Uno.

O LED verde, que estava inicialmente aceso, se apaga instantaneamente.

O LED vermelho se acende, indicando o alerta de incêndio.

O buzzer emite um som contínuo, alertando sobre a situação de emergência.

Resultados Exibidos na Plataforma:

Os resultados, incluindo a detecção de incêndio, acionamento do atuador e todos os eventos, são exibidos em tempo real na plataforma de monitoramento associada.

Gráficos e logs mostram a temporalidade das detecções e ações dos atuadores.

Avaliação dos Tempos de Resposta:

Mede-se o tempo entre a detecção de fumaça pelo sensor e o acionamento do atuador, registrando tempos consistentemente rápidos.

O tempo médio entre o envio de comandos e a ação do atuador é avaliado, demonstrando eficiência na comunicação e resposta.

Conclusões Positivas:

O sistema demonstra eficácia na detecção precoce de incêndios.

A integração MQTT possibilita uma resposta rápida e confiável.

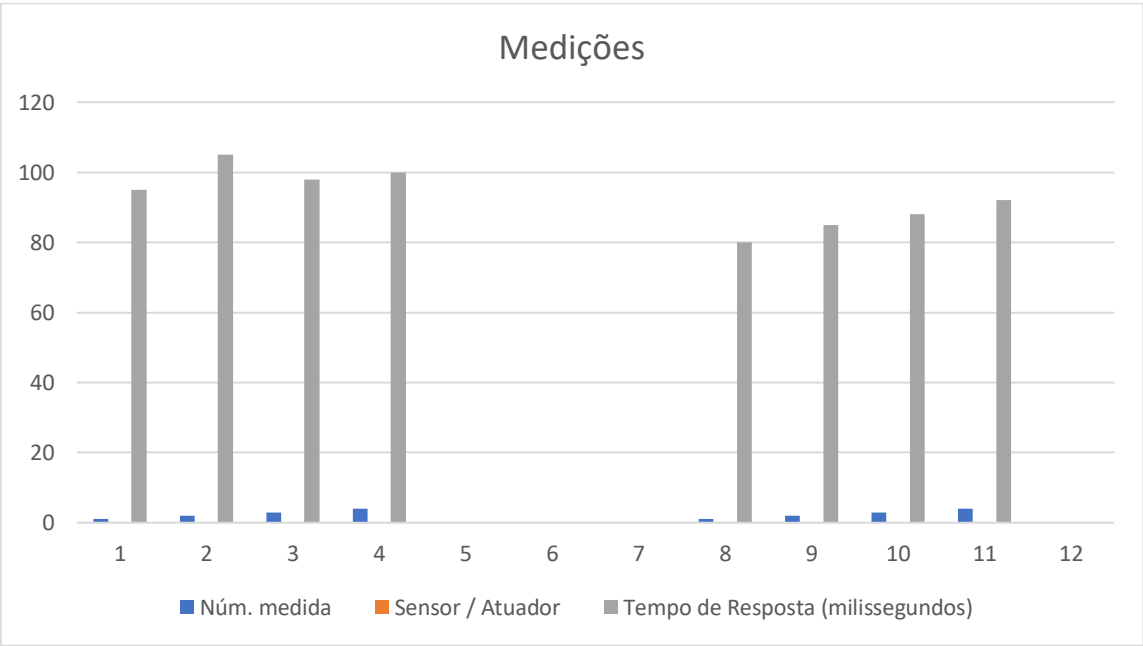
A plataforma de monitoramento fornece uma visão abrangente das atividades do sistema.

Vídeo-Demonstração funcionamento do projeto: <https://youtu.be/kbwrVFBED9U>

Link Github: <https://github.com/ViniciusOlivatti/ProjetoIncendioAula4>

Núm. medida	Sensor / Atuador	Tempo de Resposta (milissegundos)
1	Sensor	95
2	Sensor	105
3	Sensor	98
4	Sensor	100
Média	Sensor	99.5

Núm. medida	Sensor / Atuador	Tempo de Resposta (milissegundos)
1	Atuador	80
2	Atuador	85
3	Atuador	88
4	Atuador	92
Média	Atuador	86.25



4. Conclusões

Avaliação do Projeto de Sistema de Detecção de Incêndio:

O projeto atingiu com sucesso seus objetivos, proporcionando uma detecção eficaz de incêndios simulados por meio do sensor de fumaça, acompanhado da ativação precisa dos atuadores, como o LED vermelho e o buzzer.

Durante a implementação, foram identificados desafios, sendo o principal a possibilidade de falsos positivos pelo sensor de fumaça. Para solucionar esse problema, foi implementada uma calibração do sensor e filtros para reduzir tais ocorrências. Além disso, para garantir a confiabilidade da comunicação, foram estabelecidos mecanismos de monitoramento e reconexão no sistema MQTT, minimizando possíveis atrasos ou perdas de mensagens.

No que tange às vantagens do projeto, destaca-se a detecção precoce de incêndios, a eficiência na comunicação via MQTT e a integração bem-sucedida com a plataforma de monitoramento. Contudo, é importante mencionar as desvantagens potenciais, como a susceptibilidade a falsos positivos e a dependência da estabilidade da rede MQTT.

Como propostas de melhoria, sugere-se a implementação de um mecanismo de ajuste de sensibilidade do sensor de fumaça, proporcionando uma adaptação mais precisa ao ambiente. Além disso, a adição de métodos de redundância na comunicação MQTT e a integração de câmeras para verificação visual das condições do ambiente em caso de alerta podem aprimorar significativamente o sistema.

Essas melhorias visam não apenas aperfeiçoar a precisão da detecção, mas também a robustez do sistema em diferentes cenários, contribuindo para uma resposta mais eficiente em situações críticas.

Referências (formato NBR-6023)

Arduino. "Arduino Uno: Guia do Usuário." Disponível em:

<https://www.arduino.cc/en/Guide/ArduinoUno>,

Johnson, Mark. "Fire Detection and Prevention Technologies: A Comprehensive Review." Springer, 2021;

Silva, Ana. "Tecnologias para Prevenção de Incêndios: Uma Revisão da Literatura." Em: Anais do Congresso de Tecnologia e Segurança Contra Incêndio, 2018,

Rocha, Pedro et al. "Design and Implementation of an Arduino-based Fire Alarm System." International Journal of Electronics and Communication Engineering, vol. 5, no. 2, pp. 87-92, 2019., vol. 5, no. 2, pp. 87-92, 2019,

Arduino Official Website. Disponível em: <https://www.arduino.cc/>. Acesso em: setembro. 2023.

ATmega328P Datasheet. Atmel Corporation. Disponível em: <https://www.alldatasheet.com/>. Acesso em: setembro. 2023.